SUB A17

WHAT IS CLAIMED IS:

1. A method for migrating a portion of a stream of data from a first storage location to a second storage location in a computer system, comprising:

identifying at least one portion of the stream of data for migration to the second storage location;

moving said at least one portion to said second storage location; and

preserving said stream's data relationships.

SUB B1

2. A method according to claim 1, wherein said first storage location and said second storage location are located on different volumes.

3. A method according to claim 1, wherein said identifying of said at least one portion for migration includes identifying said at least one portion according to pre-set criteria.

4. A method according to claim 3, wherein said identifying of said at least one portion for migration includes specifying the size of an archive unit.

5. A method according to claim 3, wherein said identifying of said at least one portion for migration includes specifying the size of a region of updates.

6. A method according to claim 3, wherein said identifying of said at least one portion for migration includes specifying a memory allocation limit for the stream of data applicable to said first storage location.

7. A method according to claim 6, wherein said moving of said at least one portion is performed without exceeding said memory allocation limit.

8. A method according to claim 1, wherein the stream of data has at least one identifiable region of updates.

9. A method according to claim 1, wherein said identifying of said at least one portion for migration includes identifying a type of stream of data.

10. A method according to claim 9, wherein said type of stream of data is an append-only file.

11. A method according to claim 9, wherein said type of stream of data is a first storage block write only file.

12. A method according to claim 1, wherein said second storage location is a sequential access medium (SAM).

13. A method according to claim 1, wherein said first storage location is a local location and said second storage location is a remote location.

14. A method according to claim 1, wherein said first storage location is a non-secure storage location and said second storage location is a secure storage location.

15. A method according to claim 1, wherein said first storage location is an on-line location and said second storage location is an off-line location.

16. A method according to claim 1, wherein said preserving the data relationships of said stream includes generating metadata for description of said relationships.

17. A method according to claim 16, wherein said metadata for description of said relationships is formatted according to a scriptable interface capable of being incorporated into World Wide Web components.

18. A method according to claim 16, wherein said metadata for description of said relationships is formatted according to at least one of extensible markup language (XML), distributed component object model (DCOM) and Java.

19. A method according to claim 1, wherein the storage for said at least one portion in said first storage location is freed for use by the system after said at least one portion is moved to said second storage location.

20. A method according to claim 1, wherein said stream of data is a sparse file.

21. A computer-readable medium having computer-executable instructions for instructing a computer to perform the method recited in claim 1.

SUB A2

22. A data structure stored on a computer-readable medium for storing metadata relating to migration characteristics of a stream of data wherein at least one portion is migrated from a first storage location to a second storage location, comprising:

an identifier identifying the stream of data for which at least one portion is migrated;

data representative of the storage service used in connection with the migration of said at least one portion; and

data representative of the memory mappings of said at least one migrated portion.

23. A data structure stored on a computer-readable medium according to claim 22, further comprising temporal data relating to a time of migration of said at least one portion of said stream of data.

24. A data structure stored on a computer-readable medium according to claim 22, wherein said data structure is stored according to the format of a scriptable interface capable of being incorporated into World Wide Web components.

25. A data structure stored on a computer-readable medium according to claim 22, wherein said data structure is stored in at least one of extensible markup language (XML), distributed component object model (DCOM) and Java formats.

26. A data structure stored on a computer-readable medium according to claim 22, wherein said data structure is stored in a jet database.

27. A modulated data signal for carrying information that encodes a data structure as recited in claim 22.

28. An application programming interface (API) for use in a computer system, whereby a stream of data may register for administration for partial migration techniques according to the method of claim 1.

29. An API according to claim 28, whereby said interface provides a common way to generate and store metadata in connection with the partial migration of streams of data to secondary storage.

30. A computer system, comprising:

a hierarchical storage management (HSM) system for administering a stream of data for partial migration; and

a source storage location having a stream of data stored thereon being serviced by said HSM system;

wherein said HSM system identifies and migrates at least one portion of said stream of data to a target storage location according to pre-set criteria and generates metadata for the description of data relationships of said at least one migrated portion.

31. A computer system according to claim 30, wherein the HSM system specifies the size of an archive unit.

32. A computer system according to claim 30, wherein the HSM system specifies the size of a region of updates.

33. A computer system according to claim 30, wherein the HSM system specifies a memory

allocation limit for the stream of data applicable to said source storage location.

34. A computer system according to claim 33, wherein the HSM system moves at least one portion of the stream of data such that said memory allocation limit is not exceeded.

35. A computer system according to claim 30, wherein the HSM system identifies a stream of data that has at least one identifiable region of updates.

36. A computer system according to claim 30, wherein said identifying by said HSM system of said at least one portion includes identifying a type of stream of data.

37. A computer system according to claim 36, wherein said type of stream of data is an append-only file.

38. A computer system according to claim 36, wherein said type of stream of data is a first storage block write only file.

39. A computer system according to claim 30, wherein said target storage location is a sequential access medium (SAM).

40. A computer system according to claim 30, wherein said source storage location is an on-line location and said target storage location is an off-line location.

41. A computer system according to claim 30, wherein said source storage location is a non-secure storage location and said target storage location is a secure storage location.

42. A computer system according to claim 30, wherein said metadata is formatted according to a scriptable interface capable of being incorporated into World Wide Web components.

43. A computer system according to claim 30, wherein said metadata for description of said

relationships is formatted according to at least one of extensible markup language (XML), distributed component object model (DCOM) and Java.

44. A computer system according to claim 30, wherein the HSM system frees for use the storage for said at least one portion in said source storage location after said at least one portion is moved to said target storage location.

45. A computer system according to claim 30, wherein said stream of data is a sparse file.